IS4002 • 数据建模与分析基础

Homework 3 贝叶斯网络

中国科学技术大学 • 2024 春季学期

May 9, 2024

作业要求

本实验由书面部分（回答问题）和编程部分（代码填空）组成。其中：

- **编程部分：** 完整代码可在 `code` 文件夹中找到。你只需要在 submission.py 的

```
# BEGIN_YOUR_CODE

# END_YOUR_CODE
```

  部分完成代码编写。不要对除 `submission.py` 以外的文件进行更改。在完成所有代码编写后，你可以运行 `python grader.py` 得到最终分数。

  你也可以运行单个测试 (例如，1d-0-basic，更多测试样例见 `grader.py`)`python grader.py 1d-0-basic`。最后助教将依据代码的评测结果（包括隐藏测试）以及代码的正确性综合评定你的分数。

  注意： 在编程部分，建议使用 `python=3.10`。

- **书面部分：** 本次书面部分工作量较大，可以使用 markdown 或手写拍照，不再提供报告模板。你需要在提交的报告中写出所有问题的答案，包括代码。最后请提交 pdf 报告，确保工整清晰。

在完成所有题目后，你需要在 bb 系统上传一个命名为学号 _ 姓名 _HW3.zip 的压缩包，里面只用包含 submission.py 和 report.pdf 两个文件。助教会据此对你的作业进行评分。**注意事项：**

- **本次作业需独立完成，不允许任何形式的抄袭。如被发现，互相抄了一份作业的几名同学分配此作业的分数。**
- **本次作业的截止时间为** {2024/6/2}. **在此之后延迟** 1/2/3/4 **天会有** 10/30/60/100 **的额外扣分。**

如果在做作业时任何问题，可以通过 腾讯文档 向助教提问，也欢迎同学帮助解答问题。

0. 导入

0.1 自动驾驶

世界卫生组织的一项研究发现，全球每年有 124 万人死于路交通事故。为此，人们对可以精确计算的自动驾驶技术表现出了极大的兴趣。构建一个自动驾驶系统是一项极其复杂的任务。在这项作业中，你将专注于传感系统，它允许我们基于噪声传感器读数追踪其他汽车。

在这项作业中，你将运行两个文件——`grader.py` 和 `drive.py`。`drive.py` 文件不用于任何评分目的，它只是用来可视化你将要编写的代码，并帮助你观察不同方法如何导致不同的行为（并且获得乐趣！）。

让我们开始尝试手动驾驶：

```
python drive.py -l lombard -i none
```

你可以使用箭头键或‘w’、‘a’ 和‘d’ 来操控。向上键和‘w’ 会使你的车向前加速，向左键和‘a’ 会使方向盘向左转，向右键和‘d’ 会使方向盘向右转。请注意，你不能倒车或原地转弯。按‘q’ 退出。你的目标是从起点驾驶到终点（绿色框）而不发生事故。在道路上还有其他车辆，但是你不知道其他车辆的位置，你能安全地驾驶到终点吗？如果你做得不是很好也不要担心；教学人员只能 10 次中有 4 次到达终点。60% 的事故率相当糟糕，这就是为什么我们要用 AI 来做这件事。

`python drive.py` 的参数：

- -a：启用自动驾驶（与手动相对）。
- -i < 推理方法 >：使用 none, exactInference 来计算对其他汽车位置的信念分布。
- -l < 地图 >：使用此地图（例如 small 或 lombard）。默认为 small。
- -d：通过在地图上显示所有汽车来调试。
- -p：所有其他汽车保持静止（因此它们不会移动）。

0.2 问题描述

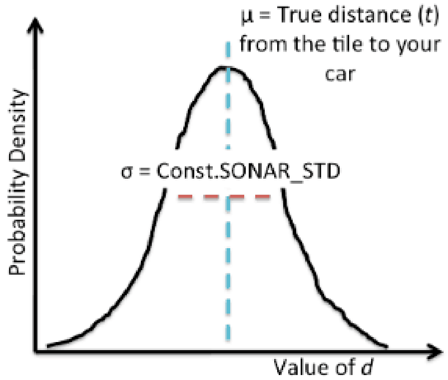
我们假设世界是一个二维矩形网格，你的车和其他 K 辆车就在这个网格上。在每一个时间步长 t，你的车会得到对每辆其他车距离的带有噪声的估计。为了简化假设，我们认为其他 K 辆车是独立移动的，对每辆车距离估计中的噪声也是独立的。因此，在下文中，我们将首先独立地对每辆其他车进行推理。在下面的符号中，我们将假设只有一辆其他车，然后再进行推广。在每一个时间步长 t，设 $C_t \in \mathbb{R}^2$ 是一对坐标，代表单个其他车辆的实际位置（未被观察到）。我们假设有一个局部条件分布 $p(c_t|c_{t-1})$ 来控制其他车辆的移动。设 $a_t \in \mathbb{R}^2$ 是你的车的位置，你可以观察并控制它。

为了降低成本，我们使用了一个基于声波的简单感应系统。该系统为我们提供了 D_t ，它是一个高斯随机变量，其均值等于你的车与另一辆车之间的真实距离，方差为 σ^2 （在代码中，σ 是 `Const.SONAR_STD`，大约是车长的三分之二）。

用数学符号表示为：

$$D_t \sim \mathcal{N}(\|a_t - C_t\|^2, \sigma^2)$$

例如，如果你的车位于 $a_t = (1, 3)$ 而另一辆车位于 $C_t = (4, 7)$ ，那么实际距离是 5，而 D_t 可能是 4.6 或 5.2 等。使用 `util.pdf(mean, std, value)` 来计算给定均值 mean 和标准差 std 的高斯概率密度函数（PDF），在 value 处进行评估。请注意，评估某个值的 PDF 不会返回一个概率——密度可以超过 1——但是为了简化，你可以将返回的 PDF 值当作概率来处理。下图显示了以你与车辆距离 $\mu = \|a_t - C_t\|_2$ 为中心的噪声距离观测 D_t 的高斯概率密度函数：



你的任务是实现一个汽车追踪器，它（大致）计算后验分布 $P(C_t|D_1 = d_1, ..., D_t = d_t)$ （你对其他车辆位置的信念），并且对每个 $t = 1, 2, ...$ 进行更新。辅助代码将负责使用这些信息来实际驾驶汽车（即，设置 a_t 以避免与 c_t 碰撞），所以你不必担心这部分。

为了简化问题，我们将世界离散化为由（row, col）对表示的网格中，其中 $0 \leq row < numRows$ 且 $0 \leq col < numCols$ 。对于每个网格，我们存储一个代表我们相信该网格上有一辆车的概率。可以通过：`self.belief.getProb(row, col)` 访问这些值。要从一个网格转换为一个位置，使用 `util.rowToY(row)` 和 `util.colToX(col)`。

0.3 贝叶斯网络

贝叶斯网络又称“信念网”，它借助有向无环图来表示变量之间的依赖关系，并使用条件概率表来描述这些变量之间的联合概率分布。在这个问题中，我们将使用贝叶斯网络来描述汽车的位置和传感器读数之间的关系。

贝叶斯网络要实现两个基本问题：

- 推断：给定一些变量的观察值，我们想要计算其他变量的后验分布。令 X 为我们想要推断的变量，E 为我们观察到的变量，已知其值为 e $= \{e_1, e_2, \ldots, e_n\}$，$P(X = x|E = e)$ 为我们想要计算的后验分布。
- 学习：给定一些数据，我们想要估计贝叶斯网络的参数。需要学习的参数包括：
  - 网络结构：变量之间的依赖关系。
  - 条件概率表：描述变量之间的联合概率分布。

1. 问题 1：概率推断 [25%]

在这个问题中，我们假设另外只有一辆车，并且它是静止的（例如，对于所有时间步 t ，$C_t = C_{t-1} = C$）。我们将表示另一辆车静止位置的单一变量记为 C，并注意 $p(C)$ 是一个已知的局部条件分布。你将实现一个函数 `observe`，该函数在观察到新的距离测量 $D_t = d_t$ 时，从当前的后验概率 $P(C|D_1 = d_1, ..., D_{t-1} = d_{t-1})$ 更新下一个时间步的后验概率 $P(C|D_1 = d_1, ..., D_t = d_t)$。当前的后验概率存储为 `ExactInference` 中的 `self.belief`。

你需要完成下面几个问题：

(a) [3 分] 画一个贝叶斯网络来描述变量 $C, D_1, D_2, D_3, a_1, a_2, a_3$ 的分布。你不需要考虑时间步长 $t = 3$ 之后的变量。

你需要回答的内容：一个贝叶斯网络的绘图，包含上述每个变量的节点。如果 $p(X|Parents(X))$ 是一个已知的局部条件分布，那么应该有一个箭头从 X 的每个父节点指向 X。

(b) [5 分] 给出联合概率 $P(C = c, D_1 = d_1, D_2 = d_2, D_3 = d_3)$ 的表达式。你的表达式应该只使用贝叶斯网络给出的已知局部条件分布。回想一下，你的车在每个时间步的位置，即 a_1, a_2, a_3，是确定已知的。

你需要回答的内容：一个数学公式，用于描述上述变量的联合概率分布，该公式仅使用问题中给出的局部条件分布。

(c) [5 分] 现在假设我们已经计算了直到时间步长 $t-1$ 的后验分布，即我们知道 $P(C|D_1 = d_1, ..., D_{t-1} = d_{t-1})$ 。我们进行了一个新的观察，即 $D_t = d_t$，我们希望将我们的后验分布更新为 $P(C|D_1 = d_1, ..., D_t = d_t)$。未归一化的更新后的后验可以计算为：

$$P(C = c|D_1 = d_1, \ldots, D_t = d_t) \propto P(C = c|D_1 = d_1, \ldots, D_{t-1} = d_{t-1})p(d_t|C = c)$$

请你证明这个公式。

你需要回答的内容：一个证明，证明上述公式的正确性。

(d) [12 分] 根据 (c) 中的公式

$$P(C = c|D_1 = d_1, \ldots, D_t = d_t) \propto P(C = c|D_1 = d_1, \ldots, D_{t-1} = d_{t-1})p(d_t|C = c)$$

在 `submission.py` 的 `ExactInference` 类中填写 `observe` 方法。这个方法应该直接修改 `self.belief`，以更新每个网格的后验概率，给定观察到的与另一辆车的噪声距离。完成后，你应该能够通过围绕它驾驶来找到静止的车（使用标志-p 意味着车辆不移动）。

注意：

- 一旦你实现了 `observe` 函数，你就可以开始使用精确推理进行驾驶。

```
python drive.py -a -p -d -k 1 -i exactInference
```

你也可以不使用 `-a` 来手动驾驶。

- 在开始之前，请阅读 `util.py` 中的 `Belief` 代码，你将需要在这次作业的几个代码任务中使用这个类。
- 记得在更新后对后验概率进行归一化处理。（在 `util.py` 中有一个有用的函数可以做到这点）。
- 在小地图上，自动驾驶车辆有时会围绕中间的障碍物转圈，然后再前往目标区域。通常，不必过于担心车辆的确切路径。相反，应专注于你的车辆追踪器是否正确地推断出其他车辆的位置。
- 如果你的车偶尔发生碰撞，也不必担心！无论是人类还是人工智能，事故都会发生。然而，即使发生了事故，你的驾驶员也应该意识到该区域有另一辆车的可能性很高。

2. 问题 2：转移概率 [25%]

现在，让我们考虑另一辆车根据转移概率 $p(C_{t+1}|C_t)$ 移动的情况。我们已经在 `self.transProb` 中为你提供了转移概率。具体来说，`self.transProb[(oldTile, newTile)]` 是已知另一辆车在时间步 t 在 `oldTile` 的前提下，在时间步 $t+1$ 在 `newTile` 的概率。

你需要完成下面几个问题：

(a) [3 分] 描述变量 $C_1, C_2, C_3, D_1, D_2, D_3, a_1, a_2, a_3$ 分布的贝叶斯网络图。你不需要考虑时间步长 $t = 3$ 之后的变量。

你需要回答的内容：一个贝叶斯网络的绘图，包含上述每个变量的节点。如果 $p(X|Parents(X))$ 是一个已知的局部条件分布，那么应该有一个箭头从 X 的每个父节点指向 X。

(b) [5 分] 给出联合概率 $P(C_1 = c_1, C_2 = c_2, C_3 = c_3, D_1 = d_1, D_2 = d_2, D_3 = d_3)$ 的表达式，该表达式仅使用已知的局部条件分布。你的表达式应该只使用贝叶斯网络给出的已知局部条件分布。回想一下，你的车在每个时间步的位置，即 a_1 a_2 a_3，是确定已知的。

你需要回答的内容：一个数学公式，用于描述上述变量的联合概率分布，该公式仅使用问题中给出的局部条件分布。

(c) [5 分] 现在假设我们已经得到了直到时间步 t 的车辆位置的条件概率 $P(C_t = c_t|D_1 = d_1, ..., D_t = d_t)$，我们已知另一辆车根据转移概率 $p(C_{t+1}|C_t)$ 移动。在时间步 t+1，我们希望计算条件概率 $P(C_{t+1} = c_{t+1}|D_1 = d_1, ..., D_t = d_t)$，证明下面的公式的正确性：

$$P(C_{t+1} = c_{t+1}|D_1 = d_1, ..., D_t = d_t) \propto \sum_{c_t} P(C_t = c_t|D_1 = d_1, ..., D_t = d_t)p(c_{t+1}|c_t)$$

你需要回答的内容：一个证明，证明上述公式的正确性。

(d) [12 分] 根据 (c) 中的公式

$$P(C_{t+1} = c_{t+1}|D_1 = d_1, ..., D_t = d_t) \propto \sum_{c_t} P(C_t = c_t|D_1 = d_1, ..., D_t = d_t)p(c_{t+1}|c_t)$$

实现一个 elapseTime 函数，该函数递归更新当前时间 t 的车辆位置的条件概率到下一个时间步 t+1。同样，后验概率存储为 ExactInference 中的 self.belief。

通过实现 elapseTime 方法完成 ExactInference。当你全部完成后，你应该能够足够好地追踪一个移动的车辆，通过运行以下命令自动驾驶：

```
python drive.py -a -d -k 1 -i exactInference
```

注意：

- 你也可以在多于一辆车的情况下自动驾驶：

```
python drive.py -a -d -k 3 -i exactInference
```

- 你也可以在 Lombard 地图上执行：

```
python drive.py -a -d -k 3 -i exactInference -l lombard
```

在 Lombard，自动驾驶可能会尝试在前往目标区域之前沿街道上下行驶。再次强调，专注于车辆追踪组件，而不是实际的驾驶过程。

3. 问题 3：是哪辆车？[30%]

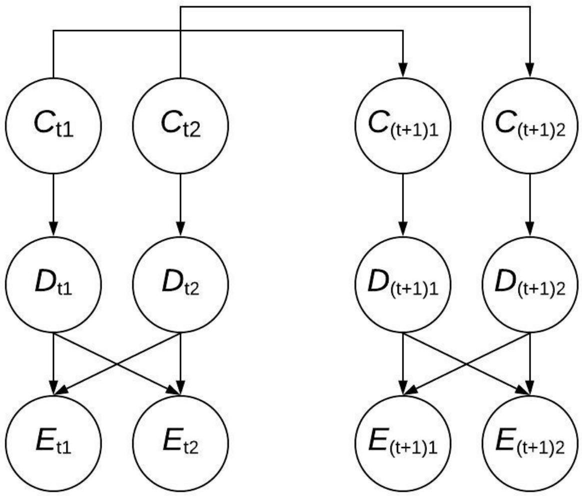
到目前为止，我们假设每辆车在每个时间步都有一个独特的噪声距离读数。实际上，我们的声波测距系统只会捕捉到这些信号的一个不加区分的集合，我们不知道哪个距离读数对应哪辆车。现在，我们将考虑这种更现实的情况。

首先，让我们扩展之前的符号：设 $C_{ti} \in \mathbb{R}^2$ 为第 i 辆车在时间步 t 的位置，其中 $i = 1, \ldots, K$，$t = 1, \ldots, T$。回想一下，所有的车辆都根据之前的转移概率独立移动。

设 $D_{ti} \in \mathbb{R}$ 为第 i 辆车在时间步 t 的噪声距离测量值，这不再是直接观察到的。相反，我们观察到的是无序的距离集合 $\{D_{t1}, \ldots, D_{tK}\}$ 作为一个整体，因此不能将这个集合中的任何测量值确定给特定的车辆。（为了简单起见，我们假设所有距离都是不同的值。）换句话说，你可以将这种情况想象为观察列表 $E_t = [E_{t1}, \ldots, E_{tK}]$，它是噪声距离测量值 $D_t = [D_{t1}, \ldots, D_{tK}]$ 的一个均匀随机排列，其中索引 i 表示时间 t 对车 i 的噪声距离。

例如，假设 $K = 2$ 和 $T = 2$。之前，我们可能在时间步 1 和 2 分别得到了第一辆车的距离读数 8 和 4，以及第二辆车的距离读数 5 和 7。现在，我们的传感器读数将是 8,5（在时间步 1）和 4,7（在时间步 2）的排列。因此，即使我们知道第二辆车在时间 $t = 1$ 时距离为 5，我们也不知道它在时间 $t = 2$ 是移动得更远（到距离 7）还是更近（到距离 4）。

下面是一个图，显示了上述情况对应的信息流，其中 $K = 2$ 并且只显示两个时间步 t 和 $t + 1$。请注意，由于观察到的距离 E_t 是真实距离 D_t 的排列，每个 E_{ti} 依赖于所有的 D_{ti}。另请注意，下图不是一个贝叶斯网络，因为 E_{t1} 和 E_{t2} 在给定 D_{t1} 和 D_{t2} 的情况下并不是条件独立的（然而，D_{t1} 和 D_{t2} 在给定 C_{t1} 和 C_{t2} 的情况下是条件独立的）。



你需要完成下面几个问题：

(a) [8 分]. 假设我们有 $K = 2$ 辆车和一个时间步长 $T = 1$。写出条件概率 $p(C_{11} = c_{11}, C_{12} = c_{12}|E_1 = e_1)$ 关于高斯概率密度函数 $p_N(v; \mu, \sigma^2)$ 和先验概率 $p(c_{11})$ 与 $p(c_{12})$ 的函数表达式。注意，对于条件 $E_1 = e_1$，我们说我们已经得到了一组观测值，但不知道哪个距离与哪辆车相关。你的最终答案不应包含变量 D_{11}, D_{12}。

注意 $p_N(v; \mu, \sigma^2)$ 是随机变量 v 在均值为 μ 和标准差为 σ 的高斯分布中的概率。

提示：对于 $K = 1$，答案将是

$$p(C_{11} = c_{11}|E_1 = e_1) \propto p(c_{11}) p_N(e_{11}; \|a_1 - c_{11}\|_2, \sigma^2)$$

其中 a_t 是你控制的车辆在时间 t 的位置。记住 C_{ti} 是第 i 辆观察到的车在时间 t 的位置。为了更好地使用贝叶斯网络，你可能会发现绘制贝叶斯网络并思考给定 $D_{t1}, \ldots, D_{tK}$ 的 E_t 分布很有用。

提示：注意观察到的变量是打乱/随机化的距离 $E_t = [E_{t1}, E_{t2}, \ldots, E_{tK}]$。这些是未观察到的噪声距离 $D_t = [D_{t1}, D_{t2}, \ldots, D_{tK}]$ 的随机排列，其中 D_{t1} 是时间步长 t 时车辆 1 的距离。注意 E_{t1} 是时间步长 t 时某辆车的测量距离，但我们不确定是哪一辆（它不一定是车辆 1，可能是任何车辆）。另一方面，D_{t1} 是车辆 1 的测量距离（我们确信它来自车辆 1），唯一的问题是我们没有直接观察到它。

提示：为了减少符号，你可以写 $p(c_{11}|e_{11})$ 替代 $p(C_{11} = c_{11}|E_{11} = e_{11})$。

你需要回答的内容：一个数学表达式，以及你推导该表达式的步骤，将 $p(C_{11} = c_{11}, C_{12} = c_{12}|E_1 = e_1)$ 与高斯概率密度函数和车辆位置的先验 $p(c_{11})$ 和 $p(c_{12})$ 相关联（可以是成比例的）。

(b) [8 分]. 假设现在你更换了传感器，使得在每个时间步 t，它们返回 K 辆车的确切位置列表，但位置列表会随机移动若干个索引（并且可以环绕）。例如，如果在时间步 1 真实的车辆位置是

$c_{11}=(1,1), c_{12}=(3,1), c_{13}=(8,1), c_{14}=(5,2)$，那么 e_1 可能是 $[(1,1),(3,1),(8,1),(5,2)]$、$[(3,1),(8,1),(5,2),(1,1)]$、$[(8,1),(5,2),(1,1),(3,1)]$ 或 $[(5,2),(1,1),(3,1),(8,1)]$，每个都有 1/4 的概率。这个移动可以从一个时间步到下一个时间步改变。

定义辅助变量 $z_1, z_2, \ldots, z_t$，它们可以用来模拟 c_t 和 e_t 之间的关系。给出 $p(c_t \mid c_{t-1})$ 关于 $e_1, \ldots, e_t$ 和 z_t 的表达式。

你需要回答的内容：辅助变量 z_t 的描述及其域，以及 $p(c_t \mid c_{t-1})$ 的表达式。

(c) [10 分]. 继问题 b 之后，描述一个有效的算法来计算任何时间步长 t 和车辆 i 的 $p(c_{ti}|e_1,\ldots,e_T)$。你的算法不应该是 K 或 T 的指数级别的。

你需要回答的内容：当建模问题时，使用的因子图/贝叶斯网络的描述，包括任何相关变量和条件概率。此外，描述你如何使用因子图来计算所提供的概率。请注意，你应尽可能简化所提供信息的概率表达式。

(d) [4 分]. 虽然对小地图进行精确推断效果很好，但它浪费了大量的计算量，因为需要为每个网格计算概率，即使对于可能没有车辆的网格也是如此。我们可以使用粒子滤波器来解决这个问题。使用粒子滤波时，程序的复杂度是 $O(\text{numParticles})$，而不是 $O(\text{numTiles})$。

要了解粒子滤波器的工作原理，请查看视频，介绍如何使用粒子滤波器来估计飞机的高度。假设我们现在使用粒子滤波，这是一种更高效的基于蒙特卡洛的汽车跟踪方法。我们已经填写了粒子滤波的代码。只需运行

```
python drive.py -a -d -k 3 -i particleFilter
```

你需要回答的内容：说明模拟与精确推理情况的不同之处，并提供原因。

4. 问题 4：模型学习 [10%]

上述的问题中，我们并没有涉及到如何学习贝叶斯网络的参数。因为贝叶斯网络的结构和概率表是已知的，所以我们可以直接进行推断。然而，在实际应用中，我们通常需要从数据中学习贝叶斯网络的参数。在这个问题中，我们将考虑如何从数据中学习贝叶斯网络的参数。

未观测的变量称为“隐变量”(latent variable)，例如，在问题 2 中，我们的隐变量是另一个车辆的位置 C_t。

在这个问题中，我们将考虑如何从观测到的数据中学习隐变量的分布，并且学习概率表的参数。我们将使用 EM 算法来学习这些参数。

EM 算法的基本思想是，我们首先初始化参数，然后交替地进行两个步骤：E 步骤和 M 步骤。在 E 步骤中，我们计算隐变量的后验分布，即给定观测数据和当前参数，隐变量的分布。在 M 步骤中，我们更新参数，以最大化对数似然。伪代码见下：

在本次课程中，我们并不希望你实现 EM 算法，但是我们希望你了解 EM 算法的基本思想。你需要完成下面几个问题：

Algorithm 1 EM 算法

Require: 观测变量 $X = x$，隐变量 Z，最大迭代次数 T

Ensure: 模型参数 $\boldsymbol{\theta}$

1: 初始化参数 $\boldsymbol{\theta}^{(0)}$

2: **for** $t = 0$ to $T - 1$ **do**

3: E **步骤**: 计算隐变量 Z 的后验分布 $q(z) = P(\mathrm{Z} = z|\mathrm{X} = x, \boldsymbol{\theta}^{(t)})$

4: M **步骤**: 最大化对数似然函数，更新参数

$$\boldsymbol{\theta}^{(t+1)} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathrm{Z}\sim q(\mathrm{Z})}[\log P(\mathrm{X}, \mathrm{Z}|\boldsymbol{\theta})]$$

5: **end for**

(a) [10 分] 考虑一个贝叶斯网络，其中包含两个二元隐变量 Z_1 和 Z_2，以及两个二元观测变量 X_1 和 X_2。隐变量之间有依赖关系，观测变量依赖于隐变量。

初始的概率表见 Table 1。

概率	值
$P(Z_1 = \text{true})$	0.6
$P(Z_1 = \text{false})$	0.4
$P(Z_2 = \text{true}\|Z_1 = \text{true})$	0.7
$P(Z_2 = \text{false}\|Z_1 = \text{true})$	0.3
$P(Z_2 = \text{true}\|Z_1 = \text{false})$	0.2
$P(Z_2 = \text{false}\|Z_1 = \text{false})$	0.8
$P(X_1 = \text{true}\|Z_1 = \text{true})$	0.9
$P(X_1 = \text{false}\|Z_1 = \text{true})$	0.1
$P(X_1 = \text{true}\|Z_1 = \text{false})$	0.5
$P(X_1 = \text{false}\|Z_1 = \text{false})$	0.5
$P(X_2 = \text{true}\|Z_2 = \text{true})$	0.3
$P(X_2 = \text{false}\|Z_2 = \text{true})$	0.7
$P(X_2 = \text{true}\|Z_2 = \text{false})$	0.6
$P(X_2 = \text{false}\|Z_2 = \text{false})$	0.4

Table 1: 初始的概率表

假设我们有观测数据集 Table 2：

X_1	X_2
true	false
true	true

Table 2: 观测数据

你需要执行一步 E、M 步骤。在 E 步骤中，你需要计算隐变量的后验分布，即给定观测数据和当前参数，隐变量的分布。在 M 步骤中，你需要更新参数，以最大化对数似然，在本题中，你可以进行对变量进行计数和归一化来最大化似然函数。

你需要回答的内容：执行一步 E、M 步骤后的概率表。

(b) [0 分] 请你阅读网页，简要了解 EM 算法的收敛性

5. Feedback [10%]

引言 你可以写下任何反馈，包括但不限于以下几个方面：课堂、作业、助教工作等等。

必填 你在本次作业花费的时间大概是？觉得难度如何？

选填 你可以随心吐槽课程不足的方面，或者给出合理的建议。若没有想法，直接忽略本小题即可。